

## Некоторые полезные строковые функции в Pandas

```
import pandas as pd
import numpy as np
```

### Функция cat

Объединяет строки в серии/индексе с заданным разделителем.

```
s = pd.Series(['a', 'b', np.nan, 'd'])
s.str.cat(sep=' ')
'a b d'
```

### Функция contains

Проверяет, содержится ли шаблон или регулярное выражение в строке серии или индекса.

```
s1 = pd.Series(['Mouse', 'dog', 'house and parrot', '23', np.NaN])
s1.str.contains('og')
0    False
1     True
2    False
3    False
4      NaN
dtype: object
```

### Функция count

Подсчитывает количество вхождений шаблона в каждой строке серии/индекса.

```
s = pd.Series(['A', 'B', 'Aaba', 'Baca', np.nan, 'CABA', 'cat'])
s.str.count('a')
0    0.0
1    0.0
2    2.0
3    2.0
4    NaN
5    0.0
6    1.0
dtype: float64
```

### Функция endswith

Проверяет, соответствует ли конец каждого строкового элемента шаблону.

```
s = pd.Series(['bat', 'bear', 'caT', np.nan])
```

```
s.str.endswith('t')
```

```
0    True
1   False
2   False
3     NaN
dtype: object
```

## Функция startswith

Проверяет, соответствует ли начало каждого строкового элемента шаблону.

```
s = pd.Series(['bat', 'Bear', 'cat', np.nan])
```

```
s.str.startswith('b')
```

```
0    True
1   False
2   False
3     NaN
dtype: object
```

*# через параметр na мы можем указать, что выводить в итоговую серию вместо NaN:*

```
s.str.startswith('b', na=False)
```

```
0    True
1   False
2   False
3   False
dtype: bool
```

```
s.str.startswith('b', na=True)
```

```
0    True
1   False
2   False
3     True
dtype: bool
```

## Функция get

Извлекает элемент из каждого элемента в указанной позиции. Извлекает элемент из списков, кортежей или строк в каждой ячейки серии/индексе.

```
s = pd.Series(["String",
               (1, 2, 3),
               ["a", "b", "c"],
               123,
               -456,
               {"1": "Hello", "2": "World"}])
```

```
s.str.get(1)
```

```
0      t
1      2
2      b
3      NaN
4      NaN
5      Hello
dtype: object
```

```
s.str.get(-1)
```

```
0      g
1      3
2      c
3      NaN
4      NaN
5      None
dtype: object
```

## Функция `isalnum`

Проверяет, все ли символы в каждой строке являются буквенно-цифровыми.

```
s1 = pd.Series(['one', 'one1', '1', ''])
```

```
s1.str.isalpha() # Проверяет, все ли символы в каждой строке являются
буквенными.
```

```
0      True
1      False
2      False
3      False
dtype: bool
```

```
s1.str.isnumeric() # Проверяет, все ли символы в каждой строке
являются числовыми.
```

```
0      False
1      False
2      True
3      False
dtype: bool
```

```
s1.str.isalnum() # isalpha + isnumeric
```

```
0      True
1      True
2      True
3      False
dtype: bool
```

## Функции islower/isupper

Проверяет, все ли символы в каждой строке являются строчными/заглавными

```
s1 = pd.Series(['leopard', 'Golden Eagle', 'SNAKE', ''])
```

```
s1.str.islower()
```

```
0    True
1   False
2   False
3   False
dtype: bool
```

```
s1.str.isupper()
```

```
0    False
1    False
2     True
3    False
dtype: bool
```

## Функция join

Соединяет значения, которые лежат в списке (list). Сами списки лежат в серии.

Если какой-либо из элементов списка не является строковым объектом, результатом соединения будет NaN.

```
s = pd.Series([['lion', 'elephant', 'zebra'],
               [1.1, 2.2, 3.3],
               ['cat', np.nan, 'dog'],
               ['cow', 4.5, 'goat'],
               ['duck', ['swan', 'fish'], 'guppy']])
```

```
s
```

```
0    [lion, elephant, zebra]
1    [1.1, 2.2, 3.3]
2    [cat, nan, dog]
3    [cow, 4.5, goat]
4    [duck, [swan, fish], guppy]
dtype: object
```

```
s.str.join('-')
```

```
0    lion-elephant-zebra
1                      NaN
2                      NaN
3                      NaN
4                      NaN
dtype: object
```

## Функция strip

Удаляет пробелы (включая символы новой строки) или набор указанных символов из каждой строки в серии/индексе с левой и правой сторон.

```
s = pd.Series(['1. Ant. ', '2. Bee!\n', '3. Cat?\t', np.nan])
```

```
s.str.strip()
```

```
0    1. Ant.  
1    2. Bee!  
2    3. Cat?  
3         NaN  
dtype: object
```

```
s.str.lstrip('123.') # lstrip удаляет только начальные символы
```

```
0    Ant.  
1    Bee!\n  
2    Cat?\t  
3         NaN  
dtype: object
```

```
s.str.rstrip('.!? \n\t') # rstrip удаляет только последние символы
```

```
0    1. Ant  
1    2. Bee  
2    3. Cat  
3         NaN  
dtype: object
```

```
s.str.strip('123.!? \n\t') # lstrip + rstrip
```

```
0    Ant  
1    Bee  
2    Cat  
3    NaN  
dtype: object
```